

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Rok Komatar

Problem pokrivnih poti

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Gašper Fijavž

Ljubljana, 2016

To delo je ponujeno pod licenco Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani www.gnu.org/licenses.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V delu obravnavajte problem pokrivnih poti na ravninskih poligonskih območjih.

Primerjajte trapezni in Boustrophedonov pristop k izračunu kratke pokrivne poti in rezultate ovrednotite na več vrstah ravninskih poligonov.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Rok Komatar sem avtor diplomskega dela z naslovom:

Problem pokrivnih poti

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Gašperja Fijavža,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 14. novembra 2016

Podpis avtorja:

Zahvaljujem se moji družini ter puncu Martyni za vzpodbudo in izkazano razumevanje tekom študija.

Prav tako se zahvaljujem programu Erasmus plus, za dano možnost študiranja na Danskem, kjer sem se prvič spoznal s temo diplomskega dela.

Nenazadnje se lepo zahvaljujem tudi prof. dr. Gašperju Fijavžu za mentorstvo in koristne nasvete pri nastajanju diplomskega dela.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Območja in tlakovanja	5
2.1	Trapezno tlakovanje	8
2.2	Boustrophedonovo tlakovanje	13
3	Pokrivanje območja	17
3.1	Enostavne pokrivne poti	17
3.2	Obhod celic	21
3.3	Najboljša smer	25
4	Rezultati	27
5	Zaključek	37

KAZALO

Slike

2.1	Območja v ravnini.	5
2.2	Tipi poligonskih črt.	6
2.3	SCPL, notranjost poligona.	7
2.4	Poligonsko območje z luknjo.	8
2.5	Tlakovanje z dominami.	9
2.6	Trapezno tlakovanje poligona.	10
2.7	Primer navpičnega razreza.	11
2.8	Izračun celic ter njigova medsebojna vidnost.	12
2.9	Trapezno tlakovanje ter graf sosednosti.	13
2.10	Primerjava Trapezne ter Boustrophedonove metode tlakovanja.	14
3.1	Ilustracija enostavnih pokrivnih poti.	18
3.2	Zadnji korak pokrivanja celice.	20
3.3	Možne poti v celicah brez leve ter desne meje.	22
3.4	Najkrajša pot v poligonu z luknjami.	23
3.5	Optimizacija globalne ter lokalne poti.	25
3.6	Rotacija poligona za 90°	26
4.1	Različna ravninska poligonska območja.	27
4.2	Predstavitev n -ogona z enotskim mrežnim n -ogonom.	28
4.3	Metoda <i>Razpihni</i>	29
4.4	Metoda <i>Reži</i>	29
4.5	Poligon z luknjami.	29

4.6	Primerjava trapeznega ter Boustrophedonovega pristopa po dolžinah poti.	30
4.7	Deli območja, ki jih ne pokrivamo.	31
4.8	Dolžine pokrivnih poti v odvisnosti od smeri tlakovanja (med 0° do 45°) na primeru kvadrata.	32
4.9	Primerjava trapeznega ter Boustrophedonovega pristopa po pokritju površine.	34
4.10	Odvisnost dolžine lokalne poti od d	35
4.11	Dolžina pokrivne poti v odvisnosti od d	35
4.12	Poligon z luknjami, ter optimiziran rdeč obhod celic.	36

Seznam uporabljenih kratic

kratica	angleško	slovensko
CPP	coverage path planning	problem pokrivne poti
CP	coverage path	pokrivna pot
SCPL	simple closed polygonal line	enostavna sklenjena poligonska črta

Povzetek

V nalogi obravnavamo problem pokrivnih poti. Pokrivna pot območja O je pot-krivulja P , od katere je vsaka točka območja O oddaljena največ za d , hkrati pa P poteka samo v notranjosti območja O . Iščemo pokrivno pot kar se da majhne dolžine. Omejimo se na primere s poligonskimi območji in poligonskimi pokrivnimi potmi. K problemu pristopimo na način deli in vladaj, poligonsko območje tlakujemo z manjšimi tlakovci, pokrivne poti na tlakovcih pa zlepimo v pokrivno pot celotnega območja. V nalogi opazujemo dve različni tlakovanji, trapezno in Boustrophedonovo tlakovanje, optimizacijo dolžine poti pa naredimo vzdolž različnih zaporedij pokrivanja tlakovcev in izbire ustrezne smeri tlakovanja. Metodi primerjamo na različnih tipih poligonov. Zaključimo lahko, da je Boustrophedonov pristop učinkovitejši, izbira optimalnega kota tlakovanja pa je bistveno odvisna od zveze med zahtevano bližino d in velikostjo območja.

Ključne besede: pokrivna pot, območje, ravninski poligon, tlakovanje, optimizacija poti, najboljša smer.

Abstract

The thesis considers the coverage path planning CPP problem. A coverage path of a region O is a path-curve $P \subseteq Q$, so that for every point $x \in O$ the distance between x and P is at most d . We are looking for a coverage path whose length is as small as possible. We focus on polygonal regions and also require that P is a polygonal line. We use a divide-and-conquer approach, we first tessellate our region with smaller tiles, respective coverage paths of individual tiles are later combined in a global coverage path. We limit ourselves to two different tessellations, a tessellation using trapeze tiles and the Boustrophedon tessellation. We optimize the length of the coverage path using different orderings of tiles along a coverage path as well as changing the direction of our tessellation. Considering several families of polygons we can state that choosing Boustrophedon tessellation outperforms the trapeze one, and that the choice of the optimal direction of the tessellation is highly dependent on the ratio between size of O and required resolution d .

Keywords: coverage path, region, plane polygon, tessellation, path optimization, best direction.

Poglavje 1

Uvod

Stalno iskanje krajših in hkrati učinkovitejših pokrivnih poti območja je za določena, ponavadi bolj kompleksna območja, vedno bolj pomembno. Dober primer je košnja travnika, obdanega z rožnimi gredicami. Travniki kosimo ter opazujemo pot središča kosilnice. Pri košnji pazimo, da kosimo samo znotraj travnika, saj rožnih gredic nočemo pokositi, ter, da pokosimo vso travo. Pot, ki jo pri tem dobimo je *pokrivna pot*. Problem načrtovanja košnje travnika lahko organiziramo vnaprej. Obliko travnika poznamo, je statična, pot kosilnice načrtujemo še pred samim začetkom košnje. Soroden problem košnji je pleskanje sten, kjer pazimo, da z barvo prekrijemo samo stene, ne pa tudi oken, vrat ipd. Upoštevati moramo tudi smer pleskanja, saj nam barva teče po steni navzdol. Iskanje pokrivne poti se lahko uporablja tudi pri oranju, kjer pazimo na smer oranja. Brazde ne smejo biti vzporedne s smerjo naklona klanca, saj nam posledično ob deževju voda polzi navzdol po brazdah ter odnaša prst. Podoben primer je pluzenje parkirišča, kjer moramo s plugom obvoziti celotno površino in hkrati sneg, ki ga plug odriva na stran, ne sme ostati na samem parkirišču.

Primer iz vsakodnevnega življenja je robotski sesalnik. Ta se mora za razliko od zgoraj opisanih problemov prilagajati trenutni razporeditvi pohištva v sobi. Stol, ki je bil včeraj pod mizo, je danes lahko na sredini sobe.

Pri vseh problemih moramo upoštevati dejstvo, da se premikamo samo

znotraj območja, ter, da kakšnih delov območja ne moremo pokriti, naj bo to ovira znotraj ali zunaj območja, ali pa v preozek kot, v katerega ne moremo zapeljati. Orodje, pri košnji kosilnico, izbiramo glede na lastnosti območja, ki ga želimo pokositi. Lončka s travo npr. ne bomo kosili s kosilnico, pač pa z škarjami.

Pot središča kosilnice opiše pot, ki se približa vsaki točki nekega območja in je v vsakem trenutku tudi vsebovana v tem območju. Tako pot imenujemo *pokrivna pot* (*coverage path*).

Pokrivna pot pri košnji je pot, ki jo prevozi kosilnica, da pokosi celotno območje, pri pluženju pa pot, ki jo prevozi plug, da spluži celotno območje.

Problem pokrivne poti (*CPP, coverage path planning*) želi poiskati pokrivno pot nekega območja. Zadovoljni smo s kakršnokoli pokrivno potjo, če je le kratka. Pri iskanju pokrivne poti se bomo omejili na poligonske poti, le-te znamo učinkovito opisati. V splošnem bi lahko dopuščali tudi matematično bolj zahtevne poti, a bi imeli težave že z njihovimi predstavitvami.

Problem pokrivne poti je podoben problemu pokrivajočega trgovskega potnika, eni od različic problema trgovskega potnika [2]. Problem trgovskega potnika je znan problem iz področja računalništva, kjer mora trgovec obiskati vsa mesta v določeni množici mest, pri tem prehoditi najkrajšo možno pot, ter se vrniti v izhodišče. Pri problemu pokrivajočega trgovskega potnika, mora trgovec namesto vsakega mesta obiskati tudi okolico vsakega mesta, pri CPP pa mora namesto okolice vsakega mesta obiskati vse točke okrog mesta. Problem trgovskega potnika je NP-težak [3, 4], kar pomeni, da je CPP NP-težak.

Problem kosilnice, ki išče najkrajšo pot, s katero pokosimo celoten travnik, pri tem pa dopuščamo vožnjo tudi izven travnika, je ravno tako NP-težak [2].

Obstaja več vrst pokrivnih algoritmov. Off-line algoritmi temeljijo na informacijah, ki so statične, območje je poznano vnaprej. Probleme, kjer območje ni poznano vnaprej rešujejo on-line algoritmi. Takšni algoritmi v stvarnem času osvežujejo znanje o pokrivnem območju. Naključni algoritmi

pokrivajo območje naključno toliko časa, dokler le to ni pokrito v celoti. Zelo neprimerni so za velike površine, kjer sta porabi energije ter časa občutno bolj pomembni kot pri manjših površinah.

Pristop, ki ga v diplomskem delu uporabljamo, razbije pokrivno območje na manjše dele, celice. Na vsaki od celic določi nekaj variant lokalnih pokrivnih poti, nato pa z ustreznimi prehodi med celicami sestavi globalno pokrivno pot. Pri tem za doseganje kratke globalne poti uporablja ustrezne izbire lokalnih pokrivnih poti kot tudi optimizacijo prehodov med celicami.

Za razbijanje pokrivnega območja smo uporabili dva algoritma, trapezno tlakovanje ter Boustrophedonovo tlakovanje. Slednja sta pri problemu pokrivne poti najbolj pogosta, saj delujeta na poligonskih območjih. Razlikujeta se v načinu delitve poligonskega območja v celice. Prvi uporablja večje število manjših celic, zato obravnava večje število potratnih prehodov med celicami. Drugi vnaprej združuje majhne celice v večje, število prehodov med celicami je posledično manjše.

Poleg trapezne ter Boustrophedonove metode je dobro znana tudi Morsejeva metoda [2], ki deluje na nepoligonskih območjih. Obstajajo tudi variacije vseh treh, na primer z uporabo genetskih algoritmov. Trapezna ter Boustrophedonova metoda delujeta v ravnini, poznamo pa tudi metode, ki delujejo v prostoru [2].

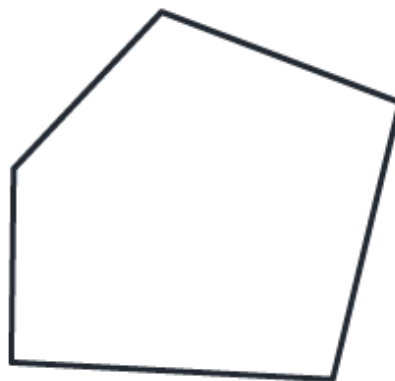
Poglavje 2

Območja in tlakovanja

Tlakovanje ravninskega območja je razbitje območja na manjša območja oziroma *tlakovce*, ki se paroma ne prekrivajo, lahko pa se dotikajo. Ravninsko območje je lahko zapleteno z zahtevnimi geometrijskimi lastnostmi, za tlakovce pa bomo želeli, da so kar se da enostavni. Pokrivno pot ravninskega območja bomo napeljali kot stik pokrivnih poti posameznih tlakovcev. Če so tlakovci enostavni, bomo za njihove pokrivne poti lahko uporabili enostavne recepte. Tlakovali bomo *poligonska območja*.



(a) Nепoligonsko območje omejeno z gladko krivuljo.



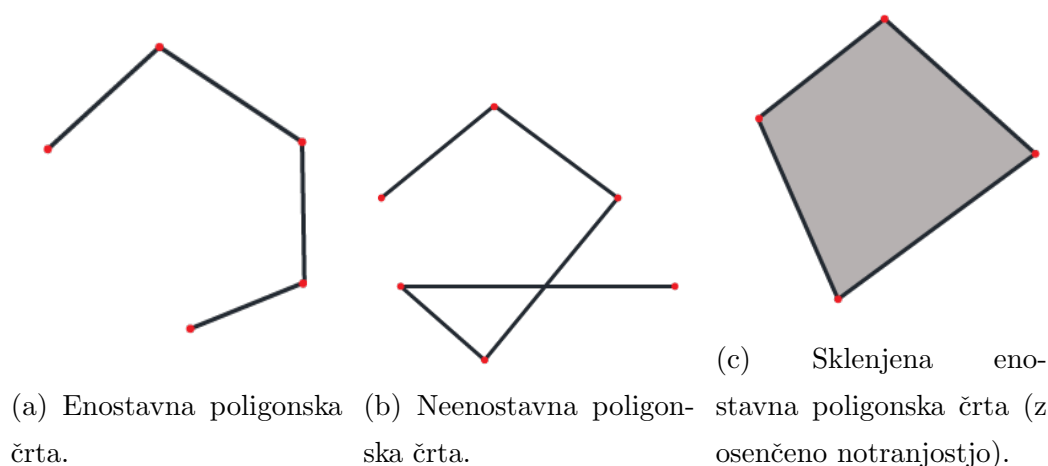
(b) Poligonsko območje omejeno s poligonsko črto.

Slika 2.1: Območja v ravnini.

Enostavna sklenjena poligonska črta (SCPL, simple closed polygonal line) je zaporedje daljic,

$$d_1, d_2, d_3, \dots, d_k \quad (2.1)$$

ki predstavlja rob poligonskega območja. Zaporedni daljici d_i, d_{i+1} , za $i \in \{1, \dots, k-1\}$ si delita krajišče.



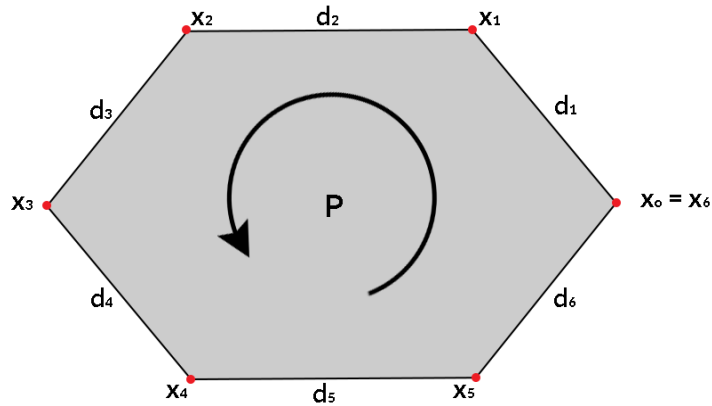
Slika 2.2: Tipi poligonskih črt.

Poligonsko črto lahko opišemo z zaporedjem točk $x_0, x_1, x_2, \dots, x_n$ v ravnini, ki porodi zaporedje daljic (2.1). Daljica d_i se začne v x_{i-1} in konča v x_i . Privzamemo lahko, da nobene tri zaporedne točke x_i, x_{i+1}, x_{i+2} ne ležijo na isti premici.

Poligonska črta (2.1) je sklenjena, če je konec zadnje daljice d_k isti kot začetek prve daljice d_1 . Glej sliko 2.2(c). Poligonska črta (2.1) je enostavna, če daljice d_1, \dots, d_k nimajo skupnih točk, razen skupnih krajišč dveh zaporednih daljic. Glej sliko 2.2(a) ter 2.2(b).

Enostavna sklenjena poligonska črta omejuje ravninsko območje, ki mu pravimo (enostaven) poligon. Dogovorimo se, da poligonske črte podajamo tako, da imajo notranjost na levi strani. Zgled je prikazan na sliki 2.3.

Če je S enostavna sklenjena poligonska črta, potem z $\text{Int}(S)$ označimo poligon, ki ga S omejuje, $\text{int}(S)$ pa naj označuje njegovo strogo notranjost: $\text{int}(S) \cap S = \emptyset$ in $\text{int}(S) \cup S = \text{Int}(S)$.



Slika 2.3: Enostavno poligonsko območje. Osenčena notranjost poligona leži levo zaporedja daljic $d_1, d_2, d_3, \dots, d_6$.

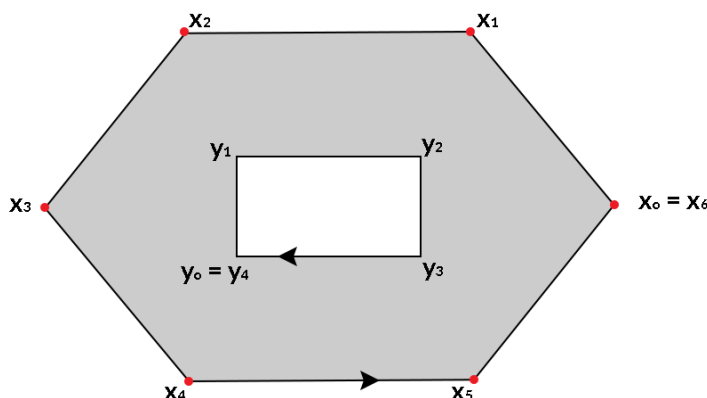
Po drugi strani, če je P (enostaven) poligon, potem z ∂P označimo njegov rob, poligonsko črto, ki omejuje P .

Definicijo poligonov in njihovih notranjosti lahko razširimo tudi na *poligone z luknjami*, ki jih opišemo z zaporedjem enostavnih sklenjenih poligonskih črt $S_0, S_1, S_2, \dots, S_k$; pri tem morajo biti poligonske črte S_2, \dots, S_k vsebovane v $\text{int}(S_1)$ in morajo imeti paroma disjunktne notranjosti.

Poligon z luknjami P_0 vsebuje natančno tiste točke v ravnini, ki pripadajo $\text{Int}(S_0)$ in ne pripadajo nobeni $\text{int}(S_i)$, za $i = 1, \dots, k$. Rob P_0 je unija vseh enostavnih sklenjenih poligonskih črt $S_0 \cup S_1 \cup \dots \cup S_k$. Zgled poligona z eno luknjo prikazuje slika 2.4.

Tlakovanje poligona P (enostavnega ali takega z luknjami) je razdelitev na poligone (*celice, tlakovce*), P_1, P_2, \dots, P_k , ki imajo paroma disjunktne notranjosti in pokrijejo celoten P . Velja $P = \bigcup_{i=1}^k P_i$. Slika 2.5 prikazuje tlakovanje šahovnice velikosti 4×4 s tlakovci v obliki domin.

Tlakovanje območja je prvi korak v iskanju pokrivne poti. Drugi korak je pokrivanje območja, torej izračun sprehoda po grafu sosednosti ter pokrivnih poti za vsako celico posebej.



Slika 2.4: Poligonsko območje z luknjo.

2.1 Trapezno tlakovanje

Konveksen štirikotnik T s stranicami a, b, c, d (po vrsti) je *trapez*, če sta a in c vzporedni. Stranici a in c imenujemo tudi *osnovnici* trapeza T , b in d sta njegova *kraka*.

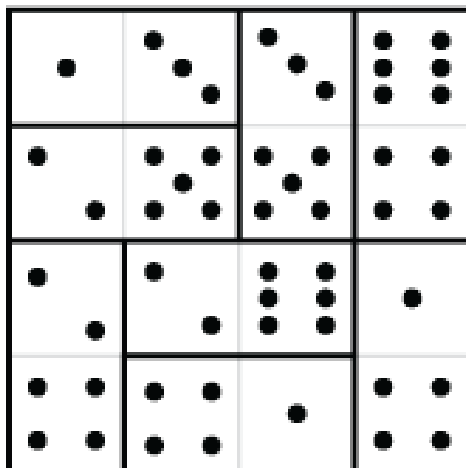
Trapeza T_1 in T_2 sta vzporedna, če so vzporedne njune osnovnice. Tudi trikotnik T' s stranicami a, b, d smemo smatrati za (izrojen) trapez, z osnovnico a , pri čemer je njegova druga osnovnica ničelna.

Trapezno tlakovanje poligona P je tlakovanje, katerega celice so vzporedni trapezi. Celice, ki jih dobimo s trapeznim tlakovanjem bomo imenovali *t-celice*.

Vsak poligon P dopušča trapezno tlakovanje. Konstrukcijo takega tlakovanja opišemo v prihodnjih treh razdelkih. Zgled trapeznega tlakovanja je predstavljen na sliki 2.6. Trapezno tlakovanje poligona P bomo dodatno opremili z grafovsko strukturo, posamezne *t-celice* bomo glede na njihovo medsebojno lego lahko smatrali za sosede v grafu.

2.1.1 Razrez

Izberimo poligon P , dopuščamo da vsebuje tudi luknje. Z X označimo množico vseh krajišč daljic njegovega roba ∂P . Brez škode za splošnost



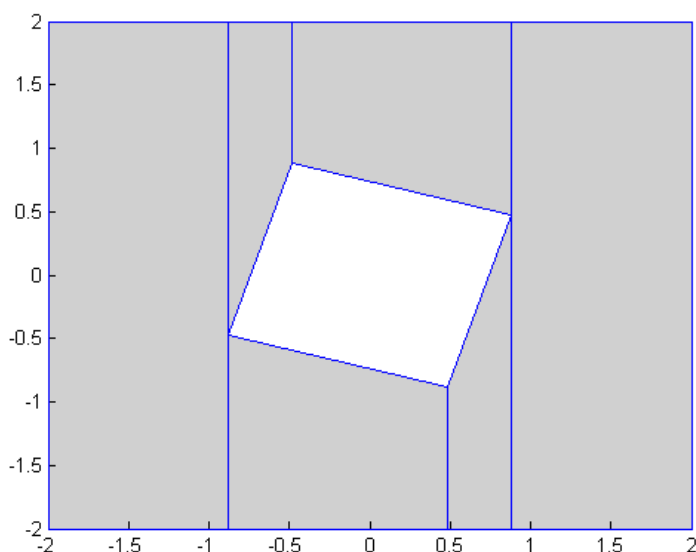
Slika 2.5: Tlakovanje z dominami.

se omejimo na primer, ko želimo poiskati tlakovanje P z navpičnimi trapezi, trapezi katerih osnovnice so navpične. Z X_1 predstavimo množico vseh x -koordinat točk iz X . Za vsako število $\alpha \in X_1$ izračunamo presek poligona P in navpične premice $x = \alpha$. Vsak tak presek je disjunktna unija segmentov/daljic, ki lahko postanejo meje trapeznih celic. Na sliki 2.7 so segmenti označeni z rdečo barvo.

2.1.2 Izračun celic

Po razrezu so celice že vidne, kako pa jih dejansko opišemo?

Najprej poiščemo levo ter desno mejo vsake celice. Predpostavimo, da je vsak segment del desne meje neke celice. Del zato, ker je lahko meja celice sestavljena iz več segmentov (na sliki 2.7 je meja pri $x = -0.9$ sestavljena iz dveh segmentov). Za vsak segment poiščemo vse levo ležeče vidne segmente. Segmenta sta medsebojno vidna, če daljica, ki povezuje središči obeh segmentov ne seka katerega od drugih segmentov ali robov območja. Ti segmenti predstavljajo levo mejo celice. Ker mejo celice lahko sestavlja več segmentov, je potrebno preveriti, če obstajajo celice, ki imajo natanko isto levo mejo, ter jih združiti.

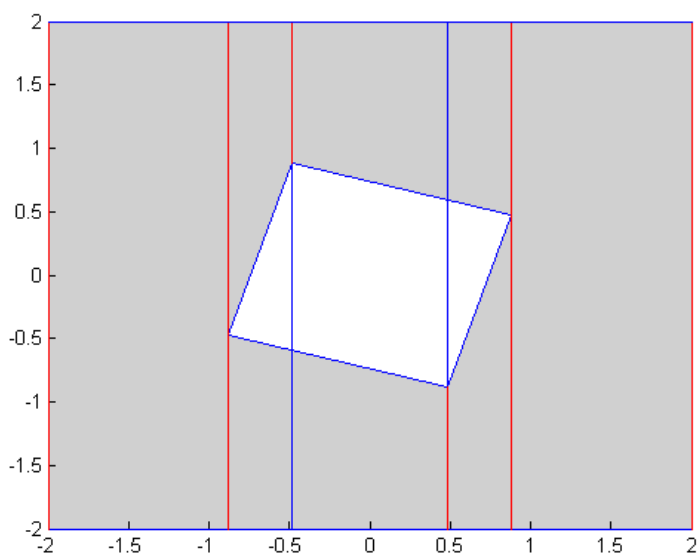


Slika 2.6: Trapezno tlakovanje poligona.

Zgled je celica št. 1 na sliki 2.8. Zelene črte predstavljajo medsebojno vidnost segmentov. Celico št. 2 na sliki 2.8, ki ima trivialno desno mejo je potrebno določiti posebej. Za izračun takšnih celic je potrebno prešteti pojavitve posameznih segmentov v mejah celic, ki jih že poznamo. Tisti segmenti, ki so se pojavili samo enkrat, predstavljajo levo mejo takšnih celic.

Izračun zgornje in spodnje meje celic je za celice, ki imajo netrivialno tako levo kot desno mejo enostaven. Zgornja meja je segment, katerega krajišči sta najvišja točka leve ter najvišja točka desne meje celice, spodnja meja pa segment, katerega krajišči sta najnižja točka leve ter najnižja točka desne meje celice.

V trapeznem tlakovanju so nekatere celice izrojene - trikotniki. Takšne celice imajo trivialno levo oziroma desno mejo. To so denimo celice z oznakami 2 ali 3 na sliki 2.8. V tem primeru je potrebno poiskati tista dva robova, ki vsebujeta zgornjo ter spodnjo točko stranske meje, ki nam je poznana. Zgornja meja takšne celice je segment od zgornje točke stranske meje do spodnje točke roba, na katerem le-ta leži, spodnja pa segment od spodnje točke



Slika 2.7: Primer navpičnega razreza. Segment, predstavljen z rdečo, je na eni strani omejen z krajiščem roba iz katerega izhajamo, na drugi pa s presečiščem navpičnice z enim od robov območja.

stranske meje do spodnje točke roba, na katerem le ta leži.

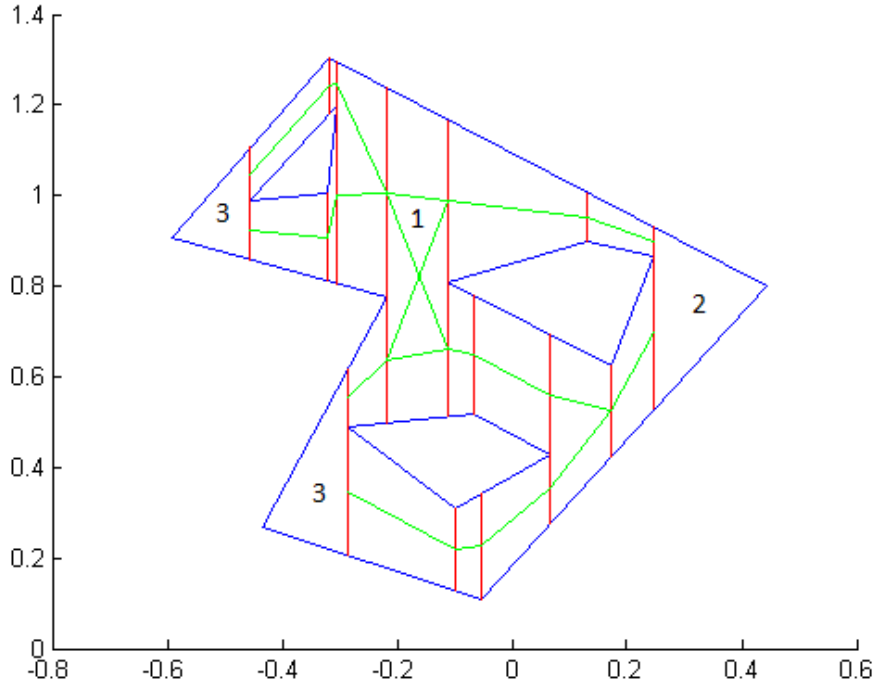
Po izračunu celic lahko zgradimo graf sosednosti.

2.1.3 Graf sosednosti celic

Graf sosednosti celic G opišemo takole. Vozlišča grafa G so celice, dve celici sta sosedni, če je presek njunih robov netrivialen (ima vsaj dve točki). Zgled grafa sosednosti celic predstavimo na sliki 2.9.

Graf sosednosti G najlažje predstavimo z matriko $M = (m_{ij})_{i,j}$ dimenzij $N \times N$, kjer N predstavlja število celic. Pri tem je $m_{ij} = 1$, če sta celici i in j sosedni, sicer je $m_{ij} = 0$.

Zapišimo matriko grafa sosednosti celic tlakovanja za prvih osem celic s

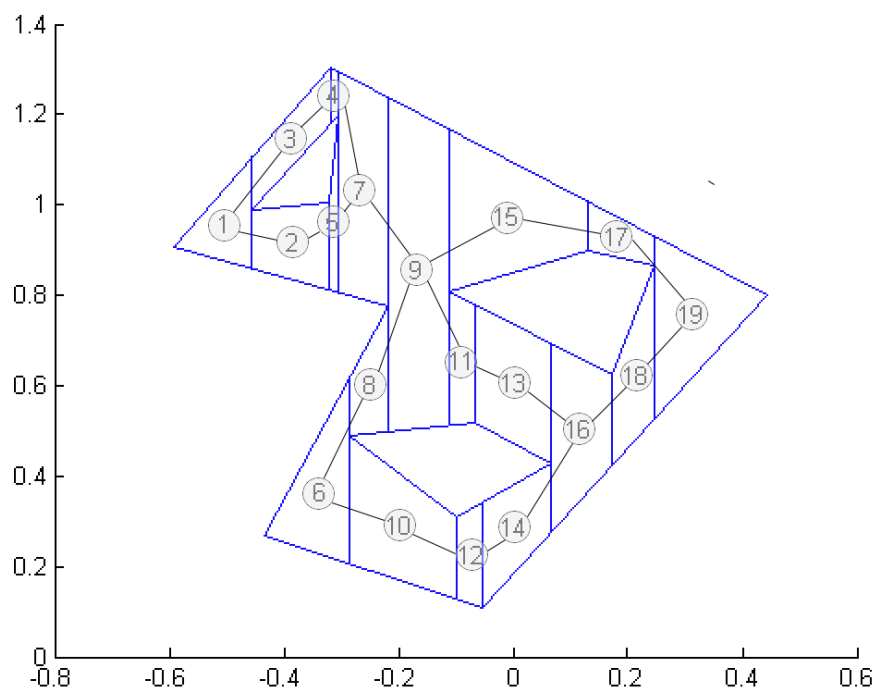


Slika 2.8: Izračun celic. Rdeče črte predstavljajo segmente, zelene pa njihovo medsebojno vidnost.

slike 2.9.

$$\begin{array}{c}
 \begin{array}{cccccccc}
 c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8
 \end{array} \\
 \begin{array}{c}
 c_1 \\
 c_2 \\
 c_3 \\
 c_4 \\
 c_5 \\
 c_6 \\
 c_7 \\
 c_8
 \end{array}
 \begin{pmatrix}
 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
 \end{pmatrix}
 \end{array}$$

Matriko sosednosti zgradimo zaporedno. Začnemo z ničelno matriko ustreznih dimenzij, nato pa za vsako celico C_i njeno levo mejo primerjamo

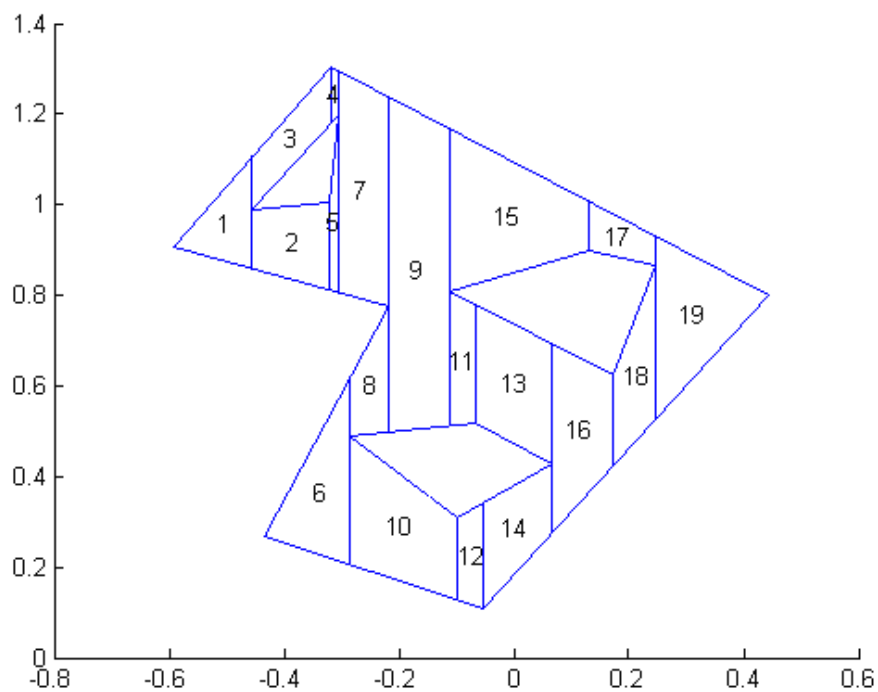


Slika 2.9: Trapezno tlakovanje ter graf sosednosti.

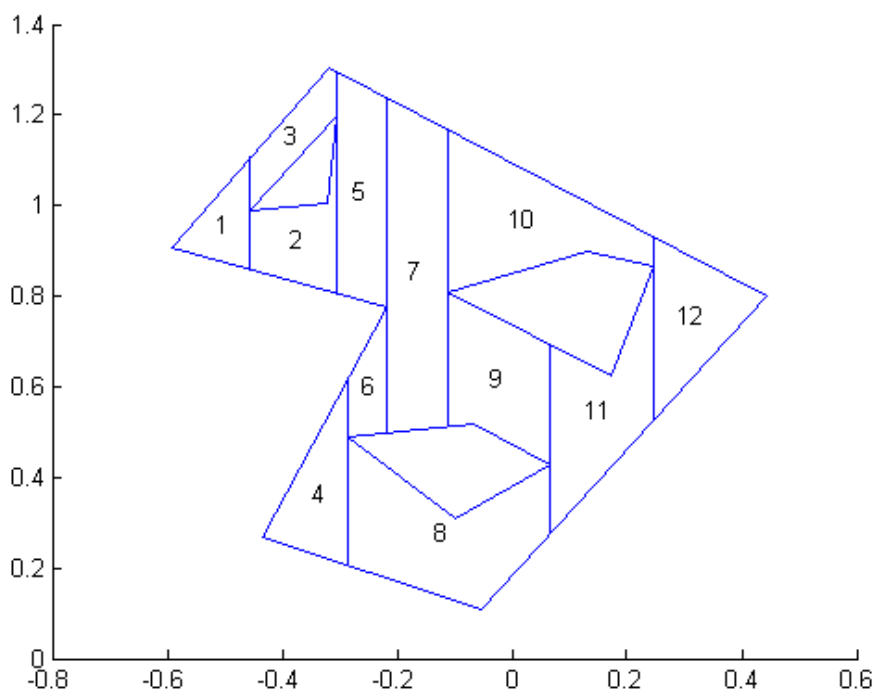
z desnimi mejami levo ležečih celic. Če imata meji celic C_i in C_j netrivialen presek, potem popravimo koeficienta m_{ij} in m_{ji} . Sosednost označimo na obeh mestih, saj je graf sosednosti neusmerjen graf.

2.2 Boustrophedonovo tlakovanje

Za razliko od trapeznega tlakovanja, ki nam območje razdeli na celice konveksnih oblik, nam *Boustrophedonovo* tlakovanje razdeli območje na manjše število večjih celic, ki niso nujno konveksne. Manjše število celic zmanjša zahtevnost optimizacije lepljenja lokalnih odsekov pokrivnih poti v celotno pokrivno pot. Boustrophedonovo metodo sta vpeljala Choset in Pignon [2]. Primerjava obeh tlakovanj je prikazana na sliki 2.10.



(a) Trapezno tlakovanje.

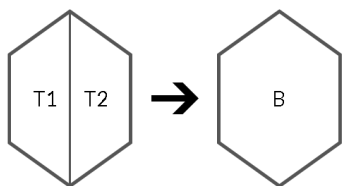


(b) Boustrophedonovo tlakovanje.

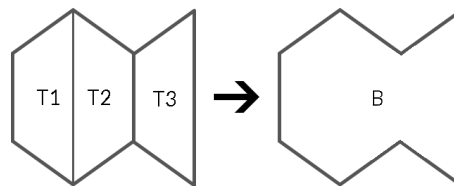
Slika 2.10: Boustrophedonovo tlakovanje z manjšim številom celic v primerjavi s trapeznim tlakovanjem.

2.2.1 Združevanje celic

Ker je Boustrophedonova metoda pravzaprav nadgradnja Trapezne metode, ponovni razrez prostora ni potreben, saj lahko ustrezne celice združimo v nove, večje celice. Rekli jim bomo tudi *b*-celice.



(a) Združitev dveh *t*-celic v *b*-celico.



(b) Združitev treh *t*-celic v *b*-celico.

Boustrophedonovo tlakovanje in ustrezen graf sosednosti *b*-celic dobimo z naslednjim postopkom.

Začnemo z trapeznim tlakovanjem T in grafom sosednosti celic G . Za vsako povezavo e v grafu G med levo celico C_i in desno celico C_j preverimo, ali je desna meja celice C_i ista kot leva meja celice C_j . Če da, potem celici C_i in C_j združimo v eno, v grafu celic pa skrčimo povezavo e . Sicer povezavo e označimo. Končamo, ko so vse povezave v grafu celic označene. Končno tlakovanje je Boustrophedonovo, ravno tako imamo na koncu izračunan graf sosednosti *b*-celic tega tlakovanja.

Smer dobljenih *b*-celic je po dogovoru identična s prvotno smerjo *t*-celic.

Poglavje 3

Pokrivanje območja

V prejšnem poglavju smo opisali, kako poligonsko območje z morebitnimi luknjami tlakujemo s celicami enostavnejših oblik. V tem poglavju bomo opisali konstrukcijo pokrivne poti. Pokrivno pot celotnega območja bomo sestavili z lepljenjem enostavnih pokrivnih poti posameznih celic.

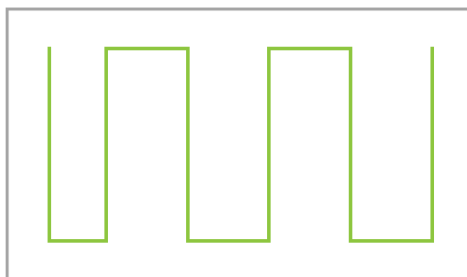
V prvem razdelku poglavja opišemo možne enostavne pokrivne poti v posamezni celici. Naslednji razdelek se ukvarja s problemom, kako pokritja posameznih celic sestavimo v pokrivno pot celotnega območja s pomočjo grafa sosednosti celic. V nadaljevanju skušamo optimizirati pokrivno pot območja z različnimi izbirami enostavnih pokrivnih poti posameznih celic in različnimi izbirami vrstnega reda obiskov posameznih celic. Poglavje zaključimo z vprašanjem, kako sploh izbrati smer celic v tlakovanju. Ideje izdelamo vzporedno za obe predstavljeni vrsti tlakovanja.

3.1 Enostavne pokrivne poti

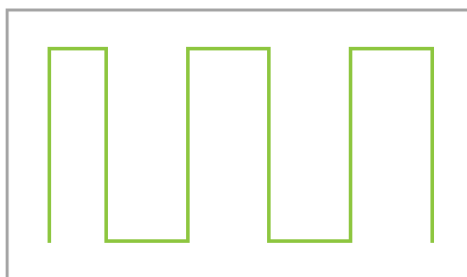
Pokrivno pot posamezne celice začenjamo blizu stranske meje. Celico lahko pokrijemo na več načinov. V primeru pravokotne pokrivne celice opišemo štiri različne enostavne pokrivne poti:

- začnemo v levem zgornjem kotu celice ter pokrivamo v desno,
- začnemo v desnem zgornjem kotu celice ter pokrivamo v levo,

- ki jih prikažemo na sliki 3.1.



(b) Začnemo desno zgoraj.



(d) Začnemo desno spodaj.

Slika 3.1: Ilustracija enostavnih pokrivnih poti v pravokotniku.

Če ima celica, denimo, samo levo mejo, desna pa je trivialna, potem bomo upoštevali zgolj dve enostavni pokrivni poti takšne celice. Obravnavamo samo primere, ko lokalno pokrivno pot začnemo bodisi levo zgoraj bodisi levo spodaj. Simetričen pristop velja za celice s samo desno mejo.

V Boustrophedonovem tlakovanju lahko pridelamo celico, ki ima trivialno tako levo kot desno mejo. Pridelamo jo takrat, ko pri tlakovanju dobimo natanko eno celico. Takšne celice pokrivamo tako, da celico na sredini delimo. Segment, ki deli celico nam služi kot začasna meja. Celico pokrijemo levo ter desno od začasne meje, nato pa poti združimo v celoto. Pri takšnih vrstah celic prav tako upoštevamo samo dva različna načina poti.

Pri prehajanju med celicami ustrezno izberemo tisto lokalno pokrivno pot, ki najmanj prispeva k dolžini celotne pokrivne poti. Začetnim ter končnim točkam pokrivne poti v posamezni celici z drugo besedo pravimo *terminali*.

3.1.1 Celice z netrivialnima stranskima mejama

Izračun lokalne pokrivne poti je za vse štiri možnosti v osnovi enak. Spodnji postopek opisuje izračun pokrivne poti celice, ki se začne v terminalu levo spodaj. Polmer kosilnice označimo z d . Implicitno prevzamemo, da so dimenzije celice bistveno večje od polmera kosilnice, in tudi, da so daljice roba celice vsaj istega velikostnega reda kot d .

Naj bo C navpična trapezna ali Boustrophedonova celica tlakovanja poligona, ki ima netrivialno tako levo kot desno mejo. Naj bosta x_L in x_R ustrezní x -koordinati. Brez škode za splošnost smemo privzeti, da je $x_L = 0$. Poleg tega naj bo k največje naravno število, za katerega velja zveza $k \cdot 2d < x_R$.

Naravno število k je izbrano tako, da lahko celico C pokrijemo s $k + 1$ vzporednimi navpičnimi pasovi, ki imajo leve x -koordinate po vrsti enake $d, 3 \cdot d, 5 \cdot d, \dots, (2k - 1) \cdot d$ in $x_R - d$.

Lokalno pokrivno pot L_C celice C bomo napeljali kot poligonsko črto, ki v vsaki drugi daljici sledi navpični simetrali zgoraj omenjenih pasov. Formalno L_C opišemo z zaporedjem točk

$$(p_i)_{i=1}^{2k+2} = (x_i, y_i)_{i=1}^{2k+2} \quad (3.1)$$

kjer koordinate x_i, y_i zadoščajo naslednjim lastnostim:

(K1) $x_i = i \cdot d$, za vsak lih indeks $i < 2k$ in $x_{2k+1} = x_R - d$,

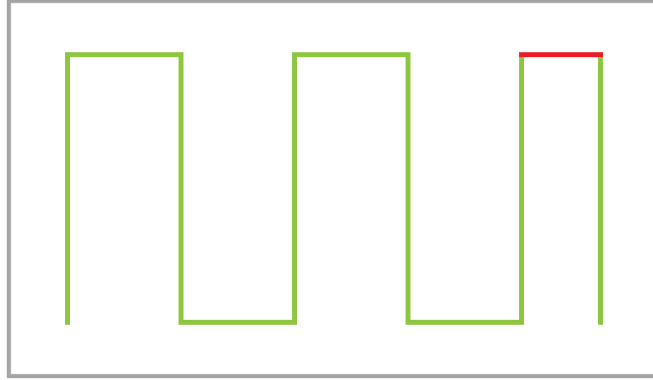
(K2) $x_i = x_{i-1}$, za vsak sod indeks i ,

(K3) če je $i \equiv 1 \pmod{4}$ ali $i \equiv 4 \pmod{4}$, je y_i najmanjše realno število, za katerega je točka (x_i, y_i) oddaljena od spodnje meje celice C za natanko d in

(K4) če je $i \equiv 2 \pmod{4}$ ali $i \equiv 3 \pmod{4}$, je y_i največje realno število, za katerega je točka (x_i, y_i) oddaljena od zgornje meje celice C za natanko d .

Z zaporedjem (3.1) opisana lokalna pokrivna pot L_C se začne v točki (x_1, y_1) in konča v točki (x_{2k+2}, y_{2k+2}) .

V primeru, ko je $x_R - x_L < 2d$, bomo z vsakim vertikalnim odsekom pokrivne poti kosili tudi zunaj celice. Ravno tako lahko s košjo po nevertikalnih odsekih na lokalni pokrivni poti kosimo zunaj.



Slika 3.2: Skrajno desni navpični pas je od prejšnjega pasu odmaknjen toliko, da ne kosimo zunaj celice (na sliki rdeče). Ima x -koordinato $x_R - d$.

3.1.2 Celice z vsaj eno trivialno stransko mejo

V celicah z vsaj eno trivialno stransko mejo bomo upoštevali samo dve različni poti. Začetni točki, terminala, sta lahko točki, ki ležita zgoraj oz. spodaj ob netrivialni stranski meji. Pokrivamo z istimi koraki kot pri celicah z netrivialnima stranskima mejama, vendar brez zadnjega koraka. Končamo, ko ugotovimo, da bo naslednji navpični pas krajši od d .

Prav tako bomo v celicah z obema trivialnima stranskima mejama upoštevali samo dve različni poti. Za prvo pot izberemo začetni točki obeh delov poti spodaj ob začasni meji, za drugo pa zgoraj ob začasni meji, kar

prikazujeta sliki 3.3(a) ter 3.3(b). Po izračunu obeh delov poti jih združimo v eno pot tako, da povežemo začetni točki obeh poti.

3.2 Obhod celic

Za pokritje celotnega območja moramo pokriti vsako celico območja, posledično potrebujemo obhod, ki obiše vse celice, vsa vozišča grafa sosednosti. Ko končamo se vrnemo nazaj v začetno celico po najkrajši možni poti. Prav tako se med celicami premikamo po najkrajši možni poti.

Začetni obhod dobimo z izračunom minimalnega vpetega drevesa, njegovo konstrukcijo opišemo v razdelku 3.2.2. Med celicami prehajamo po najkrajši možni poti od končnega terminala v trenutni celici, do ustreznega začetnega terminala v naslednji celici. Lokalno pokrivno pot ustrezno izberemo izmed možnih poti v celici tako, da je celotna pokrivna pot čim krajša.

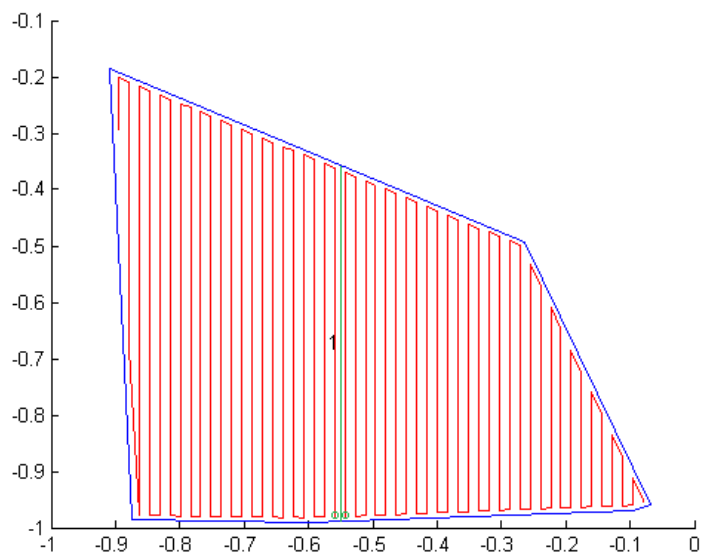
Začetni obhod optimiziramo z izbirami različnih vpetih dreves v grafu sosednosti.

3.2.1 Najkrajša pot med točkama v poligonu z luknjami

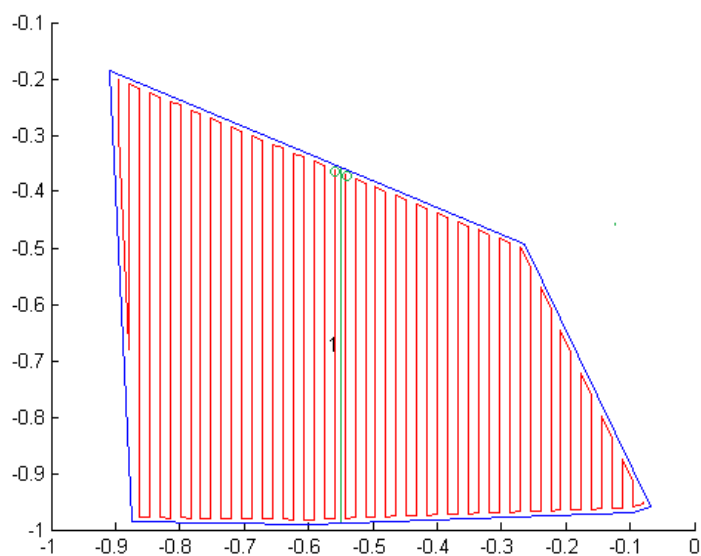
Za izračun kratke pokrivne poti potrebujemo tudi rešitev naslednjega problema. Za poligon (z luknjami) P in točki $x, y \in P$ poišči najkrajšo $x - y$ pot P_{x-y} , ki leži znotraj P . Točki x in y ležita v notranjosti poligona P .

Najkrajšo pot med točkama v poligonu lahko izračunamo s pomočjo Dijkstrovega algoritma za izračun najkrajše poti, za kar potrebujemo graf vidljivosti poligona P , $V(P)$. Graf vidljivosti $V(P)$, je utežen graf, ki predstavlja medsebojno vidljivost vseh vozlišč poligona P . V primeru, da sta vozlišči medsebojno vidni, poznamo evklidsko razdaljo med njima. Vozlišči sta medsebojno vidni takrat, ko daljica, ki ju povezuje, ne seka drugega vozlišča ali roba poligona.

Problem sta prva opisala Lozano ter Perez [5], v [1] pa je opisan postopek za izračun takšne najkrajše poti.

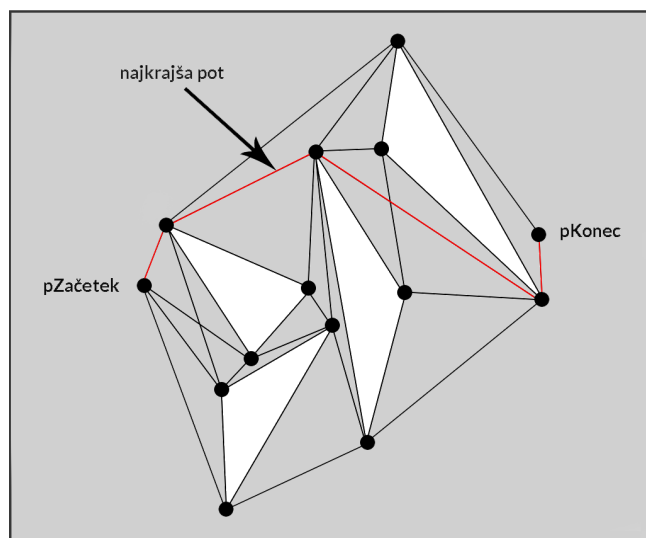


(a) Prva pot, začasna meja ter začetni točki poti.



(b) Druga pot, začasna meja ter začetni točki poti.

Slika 3.3: Začasna meja ter začetne točke poti so označene z zeleno barvo: (a) začetni točki obeh delov poti izberemo spodaj ob začasni meji, (b) začetni točki obeh delov poti izberemo zgoraj ob začasni meji.



Slika 3.4: Najkrajša pot v poligonu z luknjami.

3.2.2 Konstrukcija začetne pokrivne poti

Začetno pokrivno pot območja P bomo konstruirali na naslednji način. Najprej izberemo obhod grafa celic G , ki vsebuje vse celice C_1, \dots, C_m in se začne in konča v celici C_1 .

1. V graf G sosednosti celic tlakovanja vpeljemo dolžine povezav: dolžina povezave med sosednima celicama C_i in C_j je evklidska razdalja med njunima težiščema.
2. Naj bo T minimalno vpeto drevo grafa sosednosti celic glede na zgoraj opisano razdaljo in T^2 graf, ki ga dobimo, če vsako povezavo drevesa T podvojimo. Privzamemo, da je T korensko drevo in C_1 njegov koren.
3. Ker so točke grafa T^2 sodih stopenj lahko poiščemo Eulerjev obhod O grafa T^2 , ki se začne in konča v celici C_1 .

$$O = C_1, C_{i_2}, C_{i_3}, \dots, C_1 \quad (3.2)$$

4. Eulerjev obhod (O) reduciramo do cikličnega zaporedja celic, ki vsebuje vse celice razen C_1 natanko enkrat. C_1 je začetna in končna celica. To

lahko naredimo tako, da iz zaporedja (3.2) za vsako celico $C \neq C_1$ izbrišemo vse njene pojavitve razen prve, ravno tako izbrišemo tudi vse pojavitve celice C_1 razen skrajnih. Pridelamo zaporedje

$$C_1, C_2, C_3, \dots, C_m, C_1 \quad (3.3)$$

Pri tem smo brez škode za splošnost preimenovali celice in uskladili indekse.

5. Pokrivno pot glede na zaporedje (3.3) konstruiramo požrešno. Denimo, da smo pokrivno pot zapeljali do enega od štirih terminalov lokalnih pokrivnih poti v celici C_j . Pokrivno pot nadaljujemo z enolično določeno lokalno pokrivno potjo L_j v celici C_j in od njenega konca nadaljujemo z najkrajšo potjo $P_{j,j+1}$ do enega od terminalov lokalnih pokrivnih poti v celici C_{j+1} . V celici C_1 začnemo z najkrajšo pokrivno potjo L_1 , pot iz celice C_{m-1} pa zaključimo z najkrajšo potjo P_1 do začetka v celici C_1 .

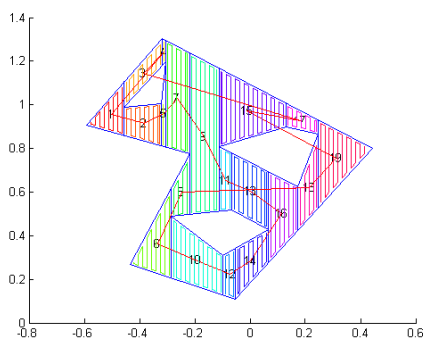
3.2.3 Lokalna optimizacija pokrivne poti

V prejšnjem razdelku smo opisali požrešno konstrukcijo pokrivne poti glede na zaporedje (3.3). Trenutno pokrivno pot smemo shematično opisati z zaporedjem

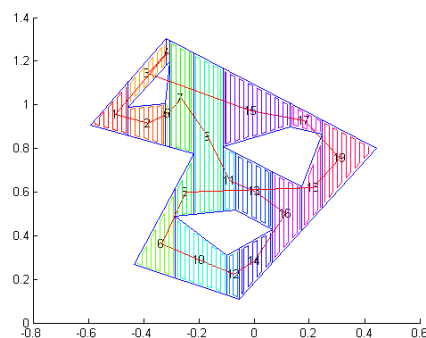
$$L_1, P_{1,2}, L_2, P_{2,3}, L_3 \dots, P_{m,1}, \quad (3.4)$$

kjer je L_i lokalna pokrivna pot celice C_i in $P_{i,i+1}$ najkrajša pot od zaključka L_i do enega od možnih začetkov lokalne pokrivne poti v C_{i+1} .

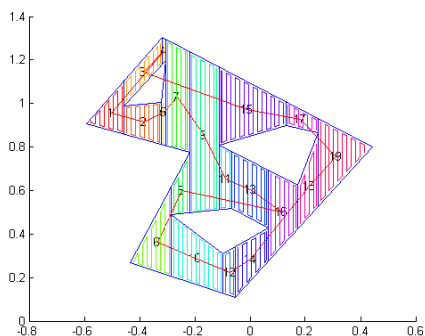
Lokalno optimizacijo modeliramo z izbirami različnih vpetih dreves v grafu sosednosti celic. Z zamenjavo ene povezave v vpetem drevesu pridemo drugačno zaporedje celic (3.3), ki potencialno porodi boljšo pokrivno pot. Lokalno optimizacijo zaključimo, ko v določenem časovnem intervalu ne uspemo najti lokalne spremembe vpetega drevesa, ki porodi boljšo pokrivno pot. Primera dveh uspešnih korakov lokalne optimizacije sta predstavljena na sliki 3.5.



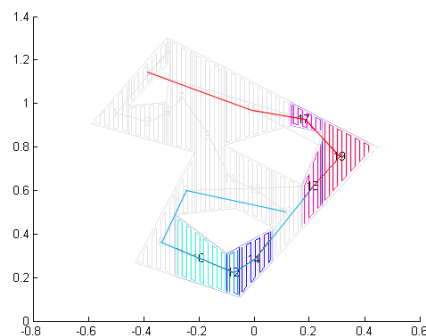
(a) Začetna pot.



(b) Prvi uspešen korak optimizacije.



(c) Drugi uspešen korak optimizacije.



(d) Razlike po optimizaciji.

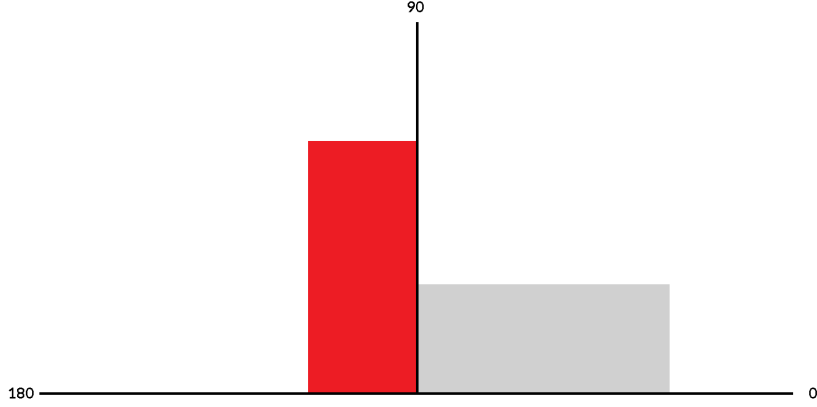
Slika 3.5: Optimizacija globalne ter lokalne poti. Rdeča pot prikazuje prehanje med celicami. Pokrivna pot posamezne celice izgleda kot obris glavnika z navpičnimi zobmi in je predstavljena v mavričnih barvah.

3.3 Najboljša smer

Do sedaj smo bodisi trapezno bodisi Boustrophedonov tlakovanje vedno konstruirali z navpičnimi trapezi oziroma b -celicami. Morda pa bi z tlakovanjem v kakšni drugi smeri uspeli najti krašjo pokrivno pot. V tem kratkem razdelku opišemo iskanje ustreznega kota tlakovanja.

Spremembo kota tlakovanja smo implementirali z rotacijo prvotnega poligonskega območja, tlakovanje pa izdelujemo v navpični smeri. Pristop je, seveda, enakovreden spremembi smeri celic za nasproten kot. Enostaven pri-

mer rotacije poligonskega območja za 90° prikazuje slika 3.6.



Slika 3.6: Rdeči poligon smo dobili z rotacijo sivega poligona za 90° okoli točke $T = \{0, 0\}$.

Optimalen kot tlakovanja poiščemo v skladu z naslednjim postopkom:

$i = 0$, $M_i = \{0^\circ, 30^\circ, \dots, 150^\circ\}$.

1. za vsak kot $\varphi \in M_i$ izračunaj dolžino pokrivne poti ℓ_φ ,
2. z $\varphi_1, \varphi_2, \varphi_3$ označimo tri kote iz M_i pri katerih so dolžine poti $\ell_{\varphi_1}, \ell_{\varphi_2}, \ell_{\varphi_3}$ najmanjše,
3. $\delta_{i+1} = \frac{1}{2}\delta$,
4. $M_{i+1} = \{\varphi_1 - \delta_{i+1}, \varphi_1, \varphi_1 + \delta_{i+1}, \varphi_2 - \delta_{i+1}, \varphi_2, \varphi_2 + \delta_{i+1}, \varphi_3 - \delta_{i+1}, \varphi_3, \varphi_3 + \delta_{i+1}\}$,
5. $i = i + 1$,

ponavljaj dokler ni $\delta_i < 1^\circ$.

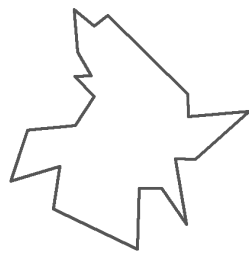
Po koncu izberemo kot z najkrajšo dolžino pokrivne poti, ki predstavlja najboljšo smer pokritja.

Podoben pristop sta za iskanje najboljše smeri uporabila Oksanen in Visala [6].

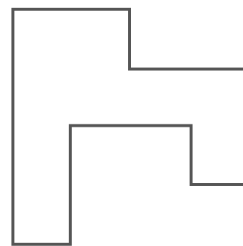
Poglavje 4

Rezultati

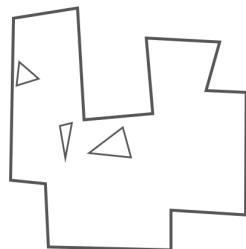
Primerjali smo trapezni in Boustrophedonov pristop k izračunu kratke pokrivne poti in rezultate ovrednotili na več vrstah ravninskih poligonov, ki jih prikazuje 4.1.



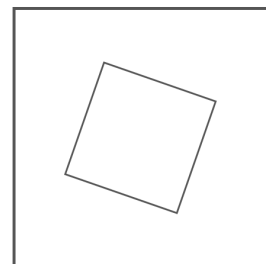
(a) Poligon zvezdaste oblike.



(b) Ortogonalen poligon.



(c) Peturbiran poligon z luknjami.



(d) Kvadrat z luknjo.

Slika 4.1: Različna ravninska poligonska območja.

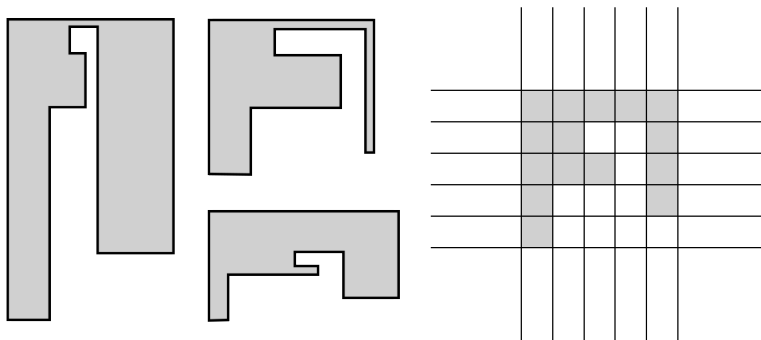
Poligon P je zvezdaste oblike, če obstaja točka s_p (rečemo ji tudi središče poligona P), da za vsako točko $x \in P$ poligon P vsebuje tudi daljico s_px [7].

Kvadrat z luknjo sestavljata dva kvadrata, zunanji s stranico dolžine 4 ter notranji s stranico $\sqrt{2}$. Notranji kvadrat je zarotiran za kot $\frac{\pi}{4} + \theta$, pri čemer θ leži na intervalu od 0 do $\frac{\pi}{2}$.

Poligon je ortogonalen, če ima vse robove navpične ali horizontalne [8]. Ortogonalne poligone smo generirali po spodnjem postopku, ki je podrobneje opisan v [8].

Ortogonalen poligon na n točkah bomo imenovali n -ogon.

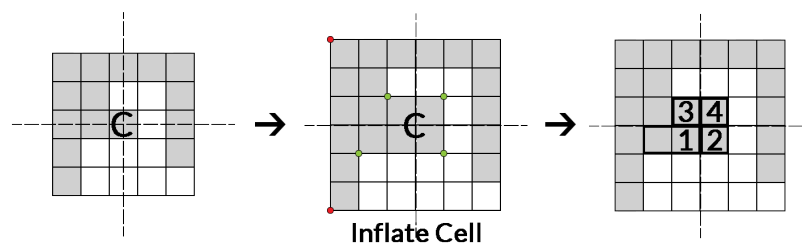
Vsak n -ogon P lahko predstavimo z mrežnim n -ogonom tako, da uredimo horizontalne robove poligona P od zgoraj navzdol, ter vertikalne robove poligona P od leve proti desni, nato pa vsakemu robu dodelimo svojo ureditveno vrednost. P tako predstavimo z enotskim mrežnim n -ogonom, glej sliko 4.2.



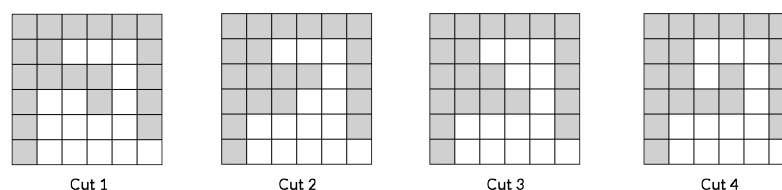
Slika 4.2: Predstavitev treh n -ogonov z enotskim mrežnim n -ogonom za $n = 12$.

Iz poligona P , predstavljenega z mrežnim n -ogonom, lahko z algoritmom *Razpihni-Reži* (*The Inflate-Cut Algorithm*) pridobimo več različnih variacij poligona P . Metoda *Razpihni* izbrano območje v mreži razširi/poveča (glej sliko 4.3), metoda *Reži* pa odstrani eno od polj mreže ter rodi nov mrežni ogon na $n+2$ točkah. Možne variacije za zgornji primer prikazuje slika 4.4.

Tudi pri generaciji Peturbiranih poligonov z luknjami smo uporabili isti

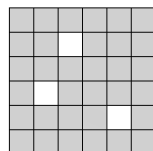


Slika 4.3: Metoda *Razpihni* poveča območje okrog kvadratka C.



Slika 4.4: Metoda *Reži* porodi več različnih $(n+2)$ -ogonov.

postopek, saj so slednji predelani ortogonalni poligoni. Kot osnovo smo vzeli ortogonalen poligon, kateremu smo dodali luknje. Enostaven primer dodajanja lukenj prikazuje slika 4.5.



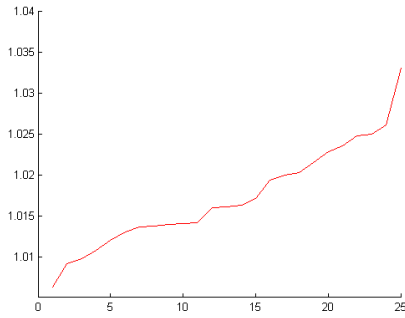
Slika 4.5: Zgornji postopek za x in y kooordinate točk izbere zaporedna naravna števila. Luknja ohranja to lastnost. Kvadrateg v poligonu (in njegov vertikalni in horizontalni pas) razpihnemo na širino tri, nato pa centralni kvadrateg odstranimo.

Ortogonalni poligon z luknjami nato še peturbiramo z naslednjim postopkom:

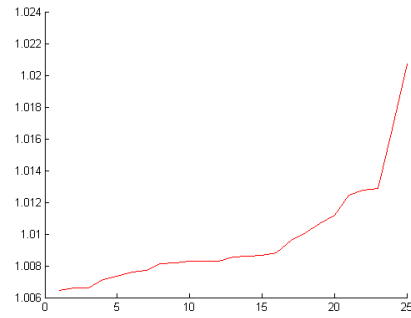
1. Izračunamo urejeni zaporedji x in y koordinat X in Y .
2. Izberemo slučajni urejeni zaporedji istih dolžin X^* , Y^* .

3. koordinato x_i (i -ti člen zaporedja X) globalno zamenjamo z x_i^* (i -ti člen zaporedja X^*).

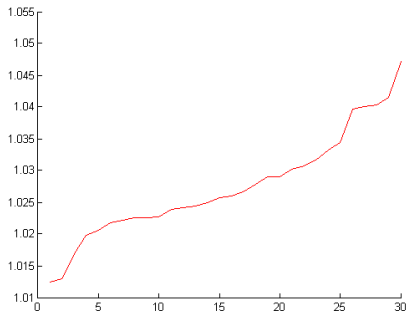
Pričakovali smo, da bo za vsako vrsto ravninskih poligonov trapezni pristop porodil daljšo, slabšo pot ℓ_T kot Boustrophedonov pristop, ki porodi pot ℓ_B . Trapezni pristop uporablja večje število celic, kar posledično ustvari večje število prehodov med celicami, ki prispevajo k končni dolžini poti. Na spodnji sliki 4.6 je za vsako vrsto ravninskih poligonov z rdečo označeno razmerje dolžin poti ℓ_T/ℓ_B . Razmerje je vedno večje od 1, kar pomeni, da je kot smo pričakovali pot ℓ_T vedno daljša od ℓ_B .



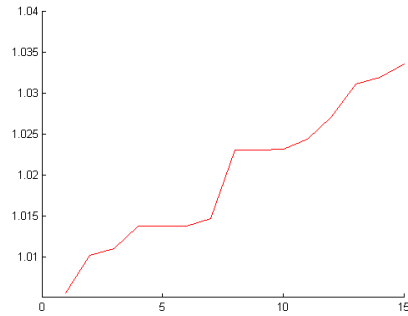
(a) Zvezdasti poligoni.



(b) Ortogonalni poligoni.



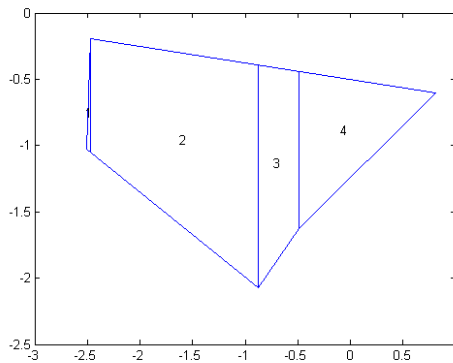
(c) Peturbirani poligoni z luknjami.



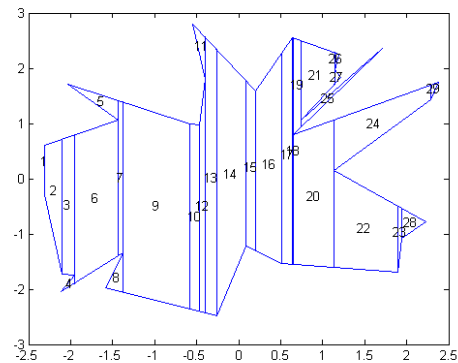
(d) Kvadrati z luknjo.

Slika 4.6: Razmerje dolžin poti ℓ_T/ℓ_B , podatki so urejeni naraščajoče. Opazimo, da je vrednost vedno večja od 1, kar pomeni, da je Boustrophedonov pristop za vse primere območij boljši, saj porodi krajšo pot.

Figura 4.9, prikazuje primerjavo razmerij $rat_T = \frac{\ell_T \cdot premerKosilnice}{povrsinaPoligona}$, ki je označeno z rdečo barvo, ter $rat_B = \frac{\ell_B \cdot premerkobilnice}{povrsinaPoligona}$, ki je označeno z modro barvo. Vrednosti razmerja 1 se hočemo čim bolj približati, saj nam predstavlja celotno površino poligonskega območja. Višek nam predstavlja dodatno površino, pokrito med prehajanjem med celicami. Vidimo, da je pri Boustrophedonovem pristopu manj prehodov, zato je tudi manj viška poti. V dveh primerih pride do primankljaja, obakrat pri zvezdastem poligonu. Do tega pride, ker moramo upoštevati dejstvo, da kakšnih delov območja ne moremo pokriti, saj kosilnica zaradi oblike ne more v preozek kot, takšna je npr. celica označena s št. 25 na sliki 4.7(b). Prav tako smo se odločili, da celic, ki so po širini manjše od polmera kosilnice ne bomo pokrivali, primer je celica označena s št. 1 na sliki 4.7(a). Takšni deli območja so najbolj pogosti pri zvezdastih poligonih, saj so najbolj razvejani.



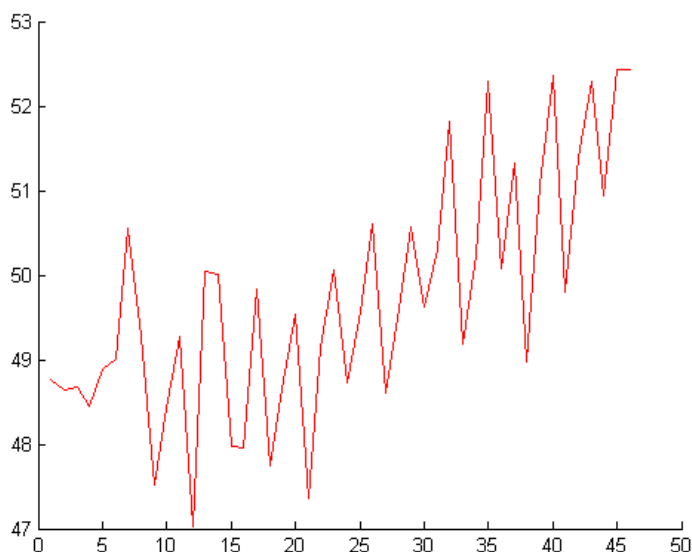
(a) Ne pokrivamo celice št. 1.



(b) Ne pokrivamo celic 1 ter 25.

Slika 4.7: Zvezdasti območji, z deli, ki jih ne pokrivamo, saj so v razmerju z d premajhni.

Pri ortogonalnih poligonskih območjih ter kvadratih z luknjo smo hoteli povedati še kaj več o optimalnem kotu, vendar rezultati niso potrdili naših domnev. Za ortogonalna poligonska območja smo domnevali, da bo najboljši kot oz. smer pokrivanja 0° , 45° , 90° , 135° ali 180° , vendar so se med optimalnimi koti pojavljali tudi drugi koti. Primer za kvadrat prikazuje slika 4.8.



Slika 4.8: Dolžine pokrivnih poti v odvisnosti od smeri tlakovanja (med 0° do 45°) na primeru kvadrata.

Pri kvadratih z luknjo smo pričakovali, da se bo smer pokrivanja prilagodila kotu zunanjega kvadrata ali pa kotu ovire, vendar se smer v veliki večini ni prilagodila ne enemmu ne drugemu.

Koti so drugačni kot smo pričakovali, ker je lokalna dolžina poti pokritja celice odvisna d , primer prikazuje slika 4.10.

Pričakovali smo, da bo pot kosilnice s premerom d pri kotu 0° krajša od poti pri kotu 45° . Če primerjamo 4.10(a) ter 4.10(c) to drži, vendar se včasih zaradi velikosti d lahko pojavi dodatna vertikala, ki podaljša pot.

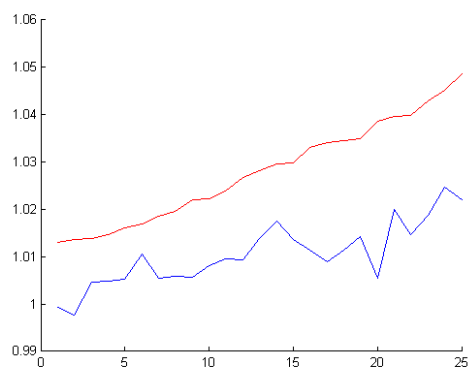
Na sliki 4.11 smo prikazali dolžino pokrivne poti v odvisnosti od d , ki smo ga zaporedoma enakomerno večali. Funkciji sta za oba kota padajoči, kar je razumljivo, saj z večjo kosilnico pokrijemo z eno potezo večjo površino, zato je tudi pot krajša. Opazimo skoke navzdol - pri zelo majhni spremembi premera kosilnice pride do znantnega padca dolžine pokrivne poti. Skoki se pojavljajo tam, kjer prihranimo vertikalo. Primer dodatne vertikale je prikazan na sliki 4.10(b), d je za malenkost manjši, zato na koncu ostane

št. menjave	trenutna dolžina poti	trenutna dolžina najkrajše poti
1	244.4483	244.4483
2	245.5927	244.4483
3	245.4876	244.4483
4	243.7058	243.7058
5	244.4483	243.7058
6	244.3811	243.7058
7	242.5125	242.5125
8	243.0012	242.5125
9	243.7058	242.5125

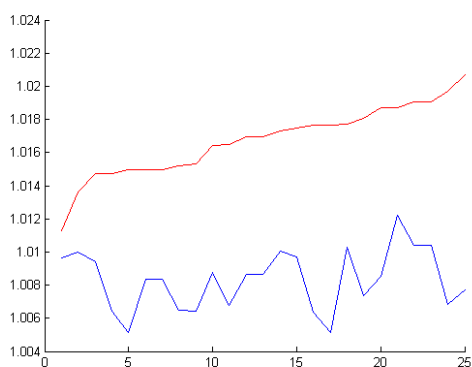
Tabela 4.1: Časovno izboljševanje pokrivne poti.

dodatna vertikalna (rdeča). Pot kosilnice z d na sliki 4.10(c) ter pot iste kosilnice na sliki 4.10(b) bi bila v tem primeru krajša.

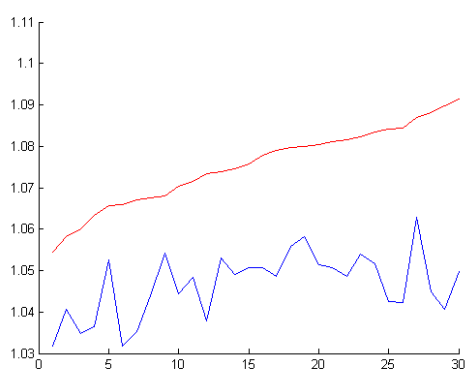
Za konec pogledjmo še časovno izboljševanje pokrivne poti, torej primer lokalne optimizacije po menjavah. Tabela 4.1 prikazuje tri izboljšave na poligonu (slika 4.12) z luknjami pri kotu 35° ter $d = 0.04$.



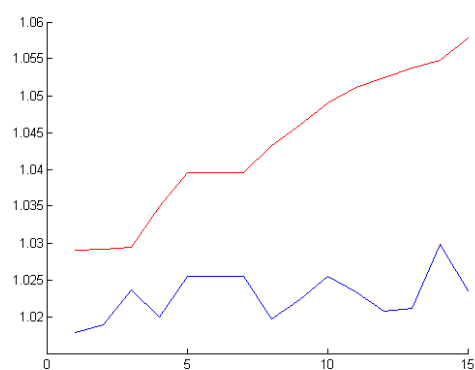
(a) Zvezdasti poligoni.



(b) Ortogonani poligoni.

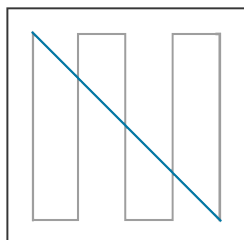


(c) Peturbirani poligoni z luknjami.

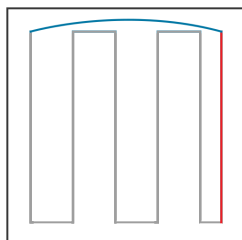


(d) Kvadrati z luknjo.

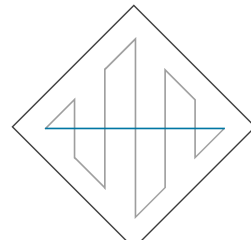
Slika 4.9: Primerjava trapeznega (rdeča) ter Boustrophedonovega pristopa (modra), glede na pokritje površine poligona. Boustrophedonov pristop je za vse primere poligonov boljši. Podatki so urejeni naraščajoče glede na trapezni pristop.



(a) Pokrivna pot pri rotaciji 0° . Ni dodatne vertikale.

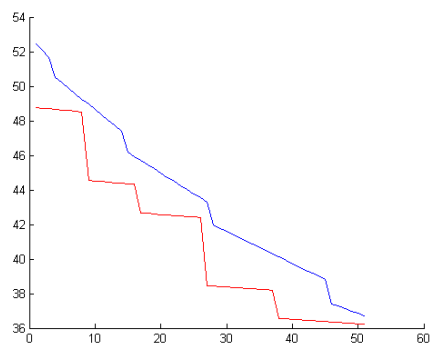


(b) Pokrivna pot pri rotaciji 0° z dodatno vertikalo.

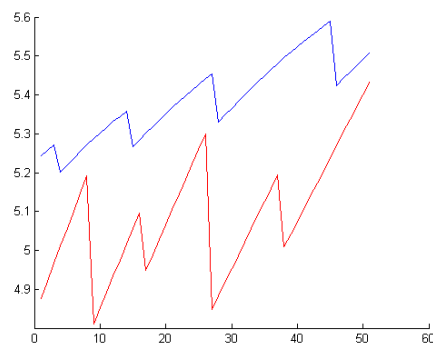


(c) Rotacija za 45° . Ni dodatne vertikale.

Slika 4.10: Odvisnost dolžine lokalne poti od d .

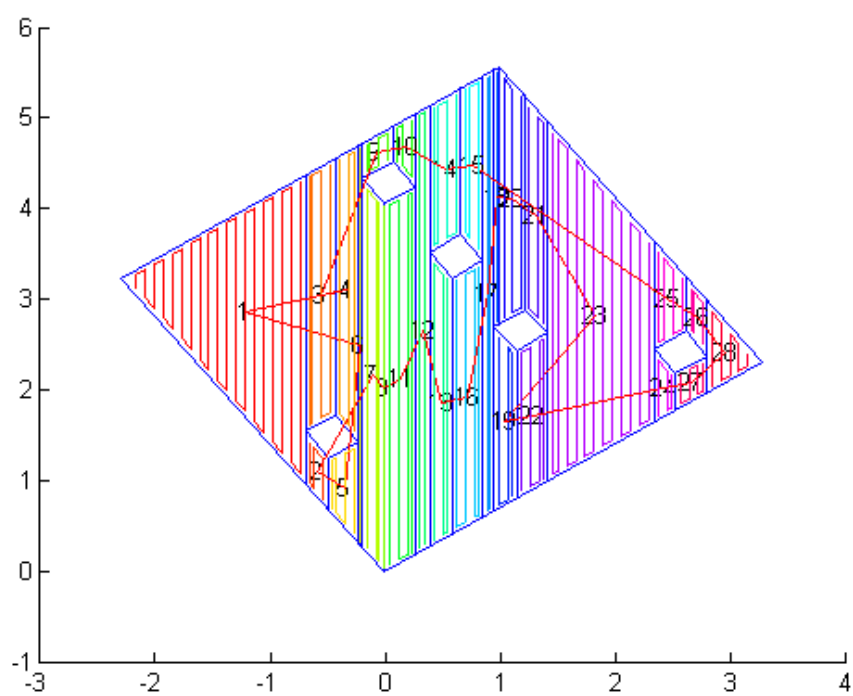


(a) Dolžina pokrivne poti.



(b) Relativna dolžina pokrivne poti.

Slika 4.11: Dolžina pokrivne poti v odvisnosti od d . Skoki se pojavljajo tam, kjer prihranimo vertikalo.



Slika 4.12: Poligon z luknjami, ter optimiziran rdeč obhod celic.

Poglavje 5

Zaključek

Primerjali smo *trapezno* in *Boustrophedonovo* metodo. *Trapezna* metoda je konsistentno slabša od *Boustrophedonove* metode. To si lahko razložimo kot posledico prehodov med celicami, ki jih ima *Boustrophedonova* metoda manj, saj združi skupine celic v eno samo, če je to možno. V povprečju smo za *Boustrophedonovo* metodo zaznali 2% izboljšanje glede na *trapezno* tlakovanje. Izboljšanje je značilnejše pri kompleksnejših poligonih, manjše pa pri poligonih z manjšim številom točk.

Za izbrani poligon P s ploščino pl in izbrano resolucijo d velja, da je dolžina pokrivne poti $\ell \geq pl/d$.

Slika 4.9 prikazuje, da sta obe metodi največ za 9% slabši od teoretičnega minimuma.

Opazili smo, da razmerje med velikostjo poligona ter premerom kosilnice vpliva na kvaliteto lokalnih pokrivnih poti v posameznih celicah. To v praksi ni težava, saj lahko izberemo najbolj primerno orodje glede na območje, ki ga želimo pokriti. Občutljivost na razmerje med velikostjo poligona in premerom pomeni, da je optimizacija dolžine pokrivne poti tako lokalni kot globalni problem. Po eni strani je potrebno iskati lokalno dobro metodo, ki v posamezni celici ni zelo slaba, po drugi strani pa je potrebno iskati tudi globalno ustrezno pokrivno pot.

Naš program bi lahko razširili tako, da bi območje tlakovali s celicami, ki

bi jih pokrivali v različnih smereh, kot sta to počela Oksanen in Visala [6]. Območje sta tlakovala s *trapezno* metodo ter izračunalna dobre smeri pokritja za vsako celico. Kot rezultat sta dobila nekaj podobnega kot mi z uporabo *Boustrophedonove* metode, vendar sta celice združevala tudi glede na najboljšo smer pokritja.

Literatura

- [1] Mark de Berg, Otfried Cheong, Marc van Kreveld in Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008.
- [2] Enric Galceran in Marc Carreras. A survey on coverage path planning for robotics. *Robot. Auton. Syst.*, 61(12):1258–1276, December 2013.
- [3] Michael Garey. *Computers and intractability : a guide to the theory of NP-completeness*. W.H. Freeman, San Francisco, 1979.
- [4] David S. Johnson in Lyle A. Mcgeoch. *The Traveling Salesman Problem: A Case Study in Local Optimization*. 1997.
- [5] Tomás Lozano-Pérez in Michael A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570, October 1979.
- [6] Timo Oksanen in Arto Visala. Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26(8):651–668, 2009.
- [7] Franco P. Preparata in Michael I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag New York, Inc., New York, NY, USA, 1985.
- [8] Ana Paula Tomás in António Leslie Bajuelos. Generating random orthogonal polygons. V Ricardo Conejo, Maite Urretavizcaya in José-

Luis Pérez de-la Cruz, *CAEPIA*, volume 3040 of *Lecture Notes in Computer Science*, str. 364–373. Springer, 2003.